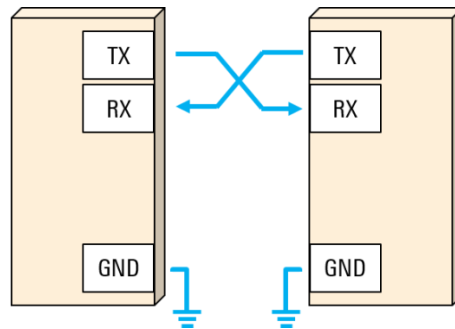


Evidencia 1

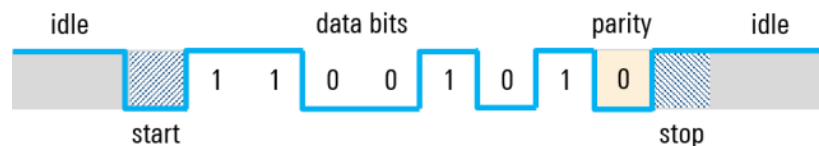
Contexto

Las siglas UART se refieren a *transmisor-receptor asíncrono universal* y lo que esto realiza es intercambiar datos en serie entre dos dispositivos. Las conexiones para este dispositivo son entre el transmisor y receptor, para poder recibir y transmitir. A continuación se puede observar un esquemático mostrando estas conexiones (Rohde & Schwarz, 2021).



(Rohde & Schwarz, 2021).

La comunicación entre estas puede ser en una sola dirección, que cada lado transmita un dato a la vez o que ambos lados transmiten en simultáneo. El formato de esta se puede observar en la siguiente imagen.



(Rohde & Schwarz, 2021).

En la imagen se pueden observar 4 estados, IDLE, START, DATA BITS y STOP. Y para que el cambio comience, se necesitan dos bits extras, o llamados de otra manera, bit de inicio y bit de parada.

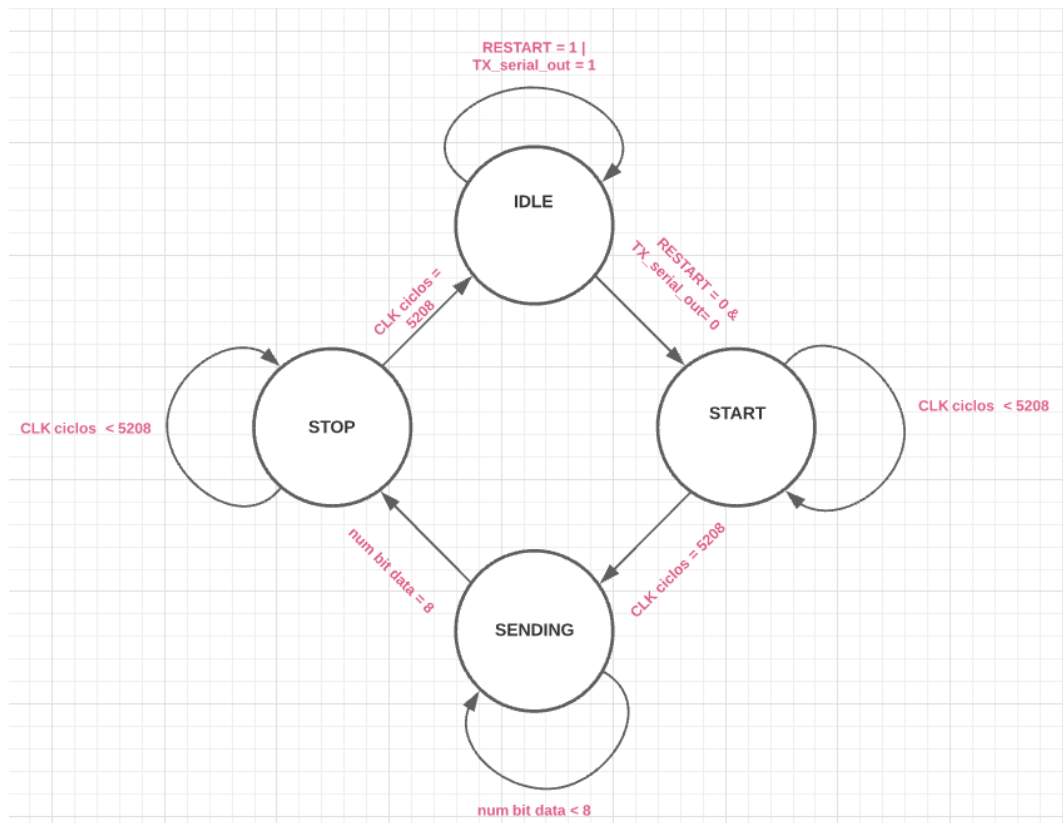
Por otro lado, es importante poner estándares en el tema de comunicación. Uno de estos es el estándar RS-232 que justamente rige a los parámetros de comunicación serial. Gracias a este protocolo las velocidades de transferencia se estandarizan, la forma de control, los voltajes utilizados, y todas las características que pueden ser utilizadas en este tipo de dispositivos (Monter, Ríos, s.f).

Análisis y proceso

Para poder crear un código VHDL, que cumpla con las especificaciones de la evidencia, se deben de tomar en cuenta el envío de bits, en este caso serán 8 bits, con un bit de inicio y otro de parada. En el desarrollo, se hicieron dos códigos, para la parte del transmisor y la parte del receptor. Cabe mencionar que a través del proceso la duración de cada bit es de 104.16 us.

Por lo tanto, si se obtienen los bits transmitidos en un segundo, se obtiene un total de 9600 bits. Por otro lado, se tiene una frecuencia igual a 50 MHz, pero no se desea esta ya que daría una duración de 0.02 us. Por lo tanto, se obtienen los ciclos de reloj por bit, dividiendo la frecuencia sobre el total de bits en un segundo. Y de resultado da un total de 5208 ciclos de reloj por bit.

Para la parte del transmisor, que lo que hace este es mandar los bits totales, tomando en cuenta la duración por bit mencionada anteriormente, 104 us. El ciclo de estados para este fue el siguiente:



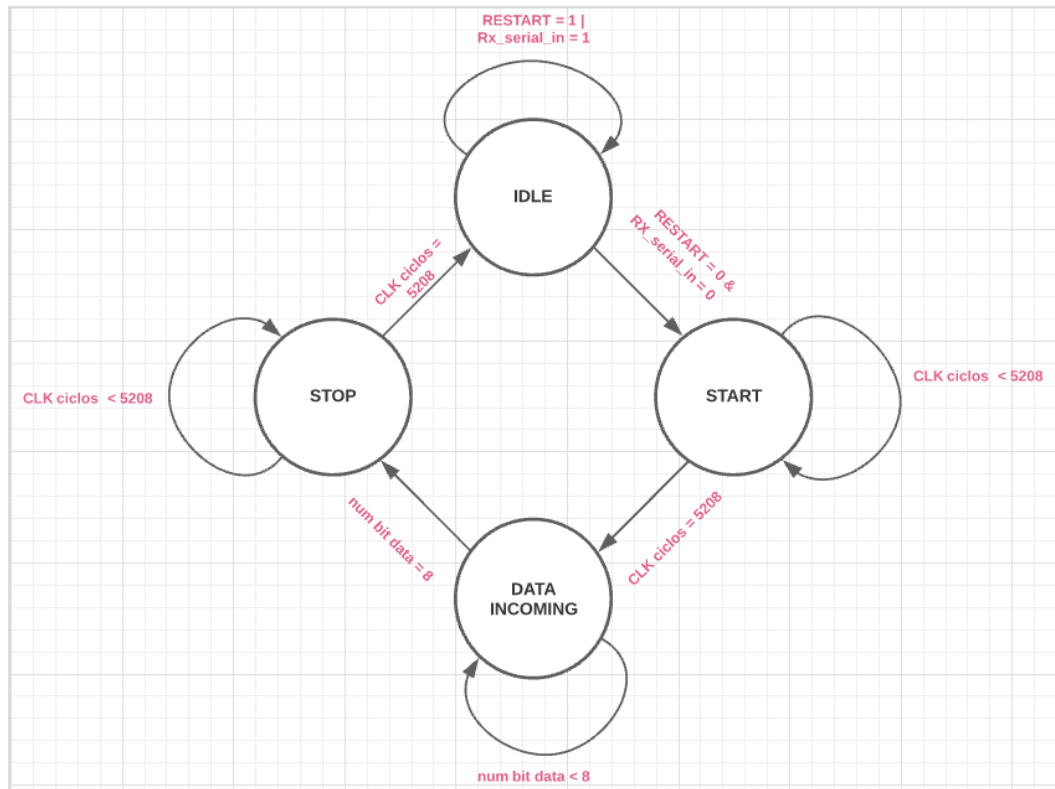
START: El transmisor comienza a mandar el bit de inicio, este estado se queda iterando hasta que los 5208 ciclos se cumplan. Esto por lo que se explicó anteriormente.

SENDING: Aquí se comienzan a mandar los 8 bits de información, esto tomando en cuenta los 5208 ciclos de reloj para cada bit. Con esto se refiere que la información que el usuario introdujo deja de ser mandada hasta que se cumplen los 5208 ciclos del octavo bit, para poder pasar al siguiente estado.

STOP: En esta etapa se manda la información completa, y se manda un 1 para que el transmisor termine de mandar información al receptor, esto siempre con la condición de cumplir los 5208 ciclos. Cabe mencionar que este 1 es indefinido hasta que llegue otra señal de entrada (0) para que reciba información nuevamente.

IDLE: Este estado se le conoce como estado de reposo, ya que mientras el usuario mande una señal de reposo o en otras palabras, no desee mandar información.. Y cuando el bit de inicio sea igual a cero, pasa al estado de inicio.

Por otra parte, para el receptor se realizó también el diagrama de estados, para poder visualizar la ruta que llevará este con el cambio de comandos.



START: Este estado es en donde el receptor sabe que empezará a recibir datos, esto gracias al bit de entrada igual a 0. Aquí también tomando los ciclos que debe tener la duración de cada bit, 5208.

SENDING: Aquí se comienzan a recibir los 8 bits de información, esto tomando en cuenta los 5208 ciclos para cada bit. Con esto se refiere que la información es recibida hasta que se cumplen los 5208 ciclos del octavo bit, para poder pasar al siguiente estado.

STOP: En esta etapa se recibe la información completa, y se recibe un 1 en donde el receptor deja de recibir información del transmisor, esto siempre con la condición de cumplir los 5208 ciclos.

IDLE: Este estado se le conoce como estado de reposo también, ya que mientras el usuario mande una señal de reposo o un bit de inicio igual a 1 y este lo reciba el receptor, se queda iterando en el estado. Y cuando el bit de inicio sea igual a cero, pasa al estado de inicio.

Desarrollo en ModelSim

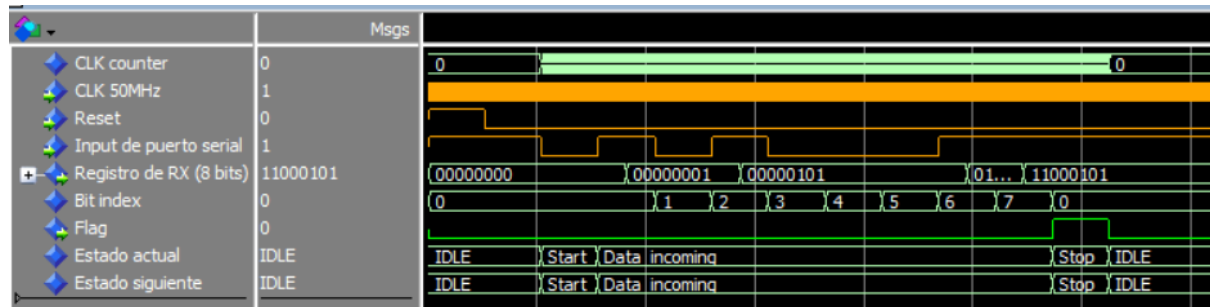
Para poder desarrollar el programa en ModelSim, se realizaron códigos encontrados en la siguiente carpeta:

[Códigos UART](#)

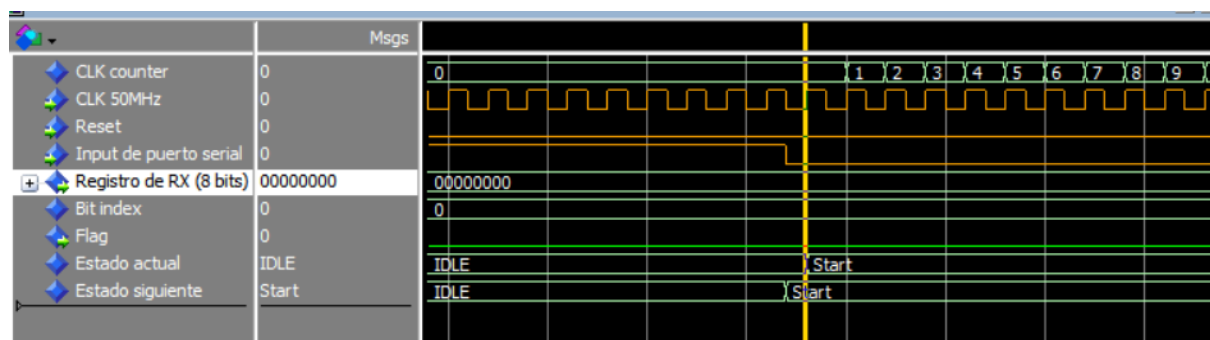
Simulaciones

Al simular los códigos presentados anteriormente, se obtuvieron los siguientes resultados.

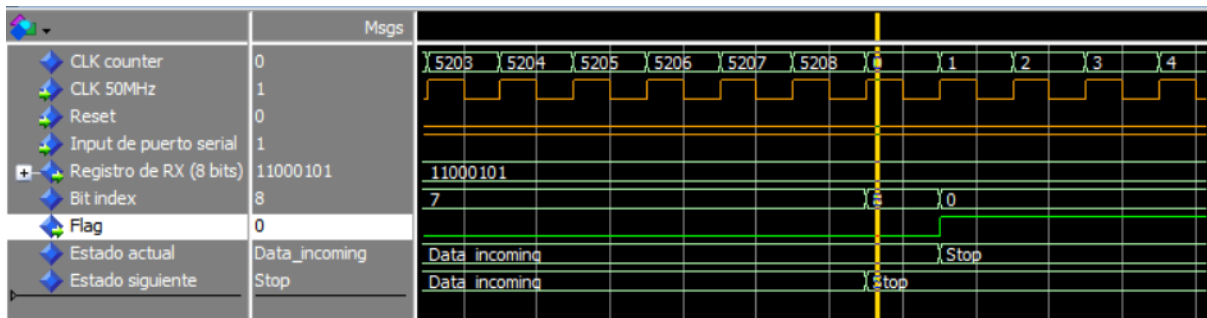
Receptor



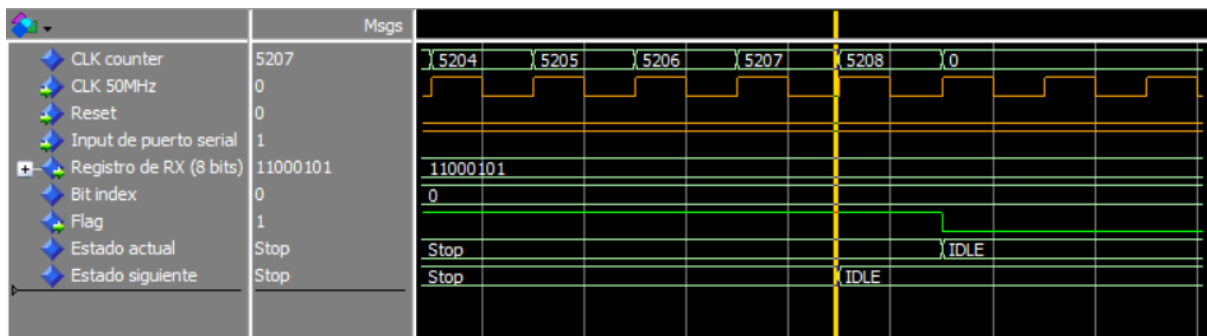
Como se puede observar, en la primera parte se tiene encendido el Reset, por lo tanto no hay cambio en los estados y se mantiene en IDLE. Cuando el Reset se apaga, se mantiene en este estado ya que el input serial se mantiene en 1 por unos momentos. Cuando este cambia a 0, el estado cambia a Start, porque es el bit de entrada. Cuando pasan los 5208 ciclos de reloj, se comienza a recibir los datos, por lo tanto, el estado cambia a data incoming. Y para que cada dato se transmita deben transcurrir 5208 ciclos. Cuando el bit index es igual a 7 (8 bits en total), comienza el bit de parada, por lo tanto cambia al estado de Stop, y también se alza la bandera que el ciclo se ha completado. Al pasar nuevamente los 5208 ciclos, el estado vuelve a su estado de reposo, o IDLE.



Por otro lado, en esta imagen, se ve más cercano el cambio en el CLK, y se puede notar que los cambios de estado se realizan cuando el CLK cambia de manera ascendente.

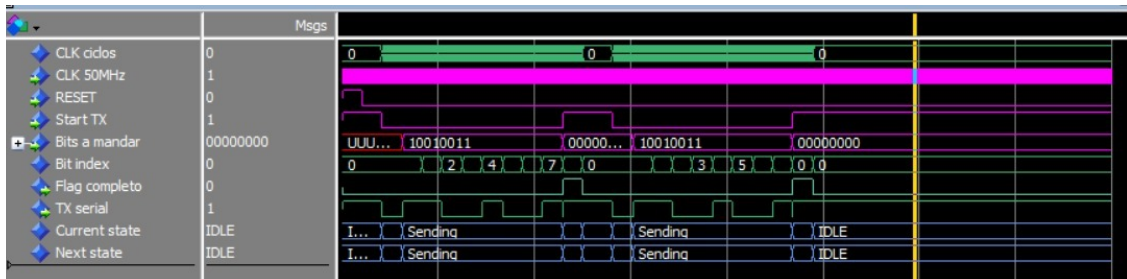


En esta imagen se puede observar la cuenta de cada ciclo, y que cuando este llega a los 5208 ciclos, el bit index cambia, para así poder recibir el siguiente bit de información. Y también se puede observar que cuando el bit index es igual a 7, pasa al bit de parada. Cabe mencionar que cuando el estado regresa a IDLE o reposo, el bit index deja de contar y se mantiene en 0.

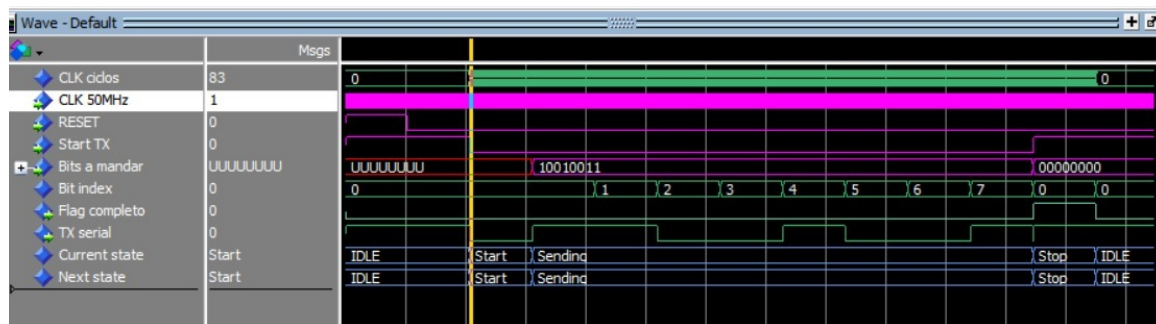


Por otra parte aquí se puede observar que cuando el bit de parada cumple con sus 5208 ciclos, el bit index se mantiene en 0, y el estado pasa a IDLE.

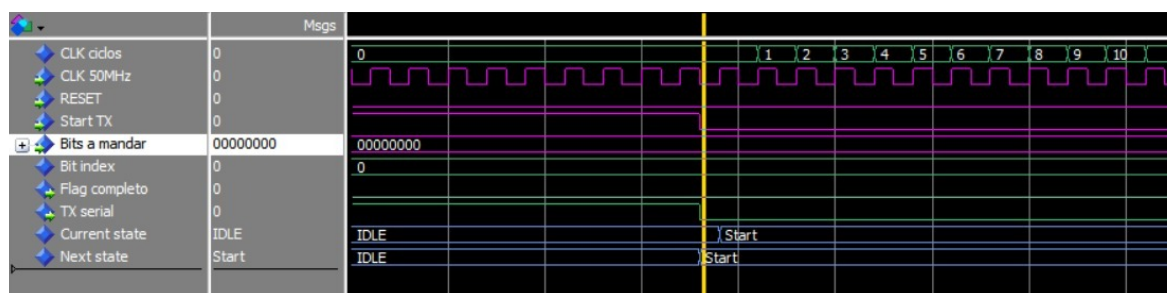
Transmisor



Para el análisis de esta segunda parte del código se comienza con esta imagen, que no se pueden apreciar bien los detalles, pero serán explicados en las siguientes imágenes. Sin embargo si se puede observar que se cumplen dos ciclos del código, ya que la bandera de completado se levanta dos veces. Además se puede observar el cambio de estados.



Para esta imagen se puede analizar los cambios de estados detalladamente. Primeramente tiene un Reset de 1, por lo tanto se mantiene en el estado de IDLE o reposo. Al apagar el Reset, se recibe la señal de inicio, por lo tanto el estado cambia a Start y comienza a leer el bit de inicio, o 0.



En esta imagen se puede observar que los cambios de estado suceden cuando el CLK cambia de manera ascendente.



Y nuevamente se puede observar el contador para los ciclos de reloj, y cuando este llega a los 5208, cambia de bit, y termina de leer cuando recibe los 8 bits y el bit de parada.

Conclusión

Debido a la buena comprensión del funcionamiento del UART, se pudo llevar a cabo la evidencia de manera correcta. Y para saber esto, se analizaron los resultados para concluir que si eran los que se esperaban.

Además, se pudo observar la eficiencia de este tipo de dispositivo para la transmisión de datos. Sin embargo, una desventaja que podría tener este es si se desean transmitir más bits de los que podría procesar.

Por otra parte, se siguió utilizando el software ModelSim que gracias a esto se puede comprender su funcionamiento cada vez mejor. También se reforzó el conocimiento de máquinas de estados, ya que al realizar el código, se tuvo en cuenta el funcionamiento de estos para que tuviera una mejor estructura y para de la misma manera, que fuera más fácil de comprender.

Referencias

L, Monter y D, Ríos. 3.4.2 Estándar RS232. Comunicaciones en redes. s.f. Recuperado de.
http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro27/342_estndar_rs232.html

Rohde & Schwarz (2021). Entendiendo el UART. Make ideas real. Recuperado de.
https://www.rohde-schwarz.com/lat/productos/prueba-y-medicion/oscilloscopios/educational-content/entendiendo-el-uart_254524.html